Skull-Closed Autonomous Development: WWN-6 Using Natural Video

Yuekai Wang, Xiaofeng Wu and Juyang Weng

Abstract—While a physical environment interacts with a human individual through the brain’s sensors and effectors, internal representations inside the skull-closed brain autonomously emerge and adapt throughout the lifetime. By “skull-closed”, we mean that the brain inside the skull is off limit to all teachers in the external physical environment, except the brain’s sensory ends and motor ends. We present the Where-What Network 6 (WWN-6), which has realized our goal of fully autonomous development inside a closed network “skull”. This means that the human programmer is not allowed to handcraft the internal representation for any fixed extra-body concepts. For example, the meanings of specific values of location or type concept are not known during the programming time. Such meanings emerge through associations imbedded in ‘postnatal’ experience. This capability is especially challenging when one considers the fact that most elements in the sensory ends are irrelevant to the signals at the effector ends (e.g., many background pixels). How does each vector value in the effector find its corresponding pattern in the correct patch of the sensory image? We outline this autonomous learning theory for the brain and present how the developmental program (DP) of WWN-6 enables the network to perform for attending and recognizing objects in complex backgrounds using natural video. The inputs to the agent (i.e., the network) are not artificially synthesized images as WWNs used before, but drawn from continuous video taken from natural settings where, in general, everything is moving.

I. INTRODUCTION

Autonomous developmental [1], [2] indicates two facts:

• The internal “brain” of the agent should be inside the skull, which is off limit to the teachers in the external environment except its sensory ends and motor ends, just like a child whose skull is closed but his eyes and arms are exposed to the external environments. This is called skull closed, which indicates that the brain develops via its interactions with sensory ends and motor ends, not the external human programmer. If it is human programmer who understands the given task and designs the brain that the agent is only supposed to perform, the agent is not developmental and its skull is open.

• The agent should be capable of learning new concepts or even tasks via the interactions with the external teacher like a child. For example, a child doesn’t know how to get dressed himself. But after taught by his parents, he will be able to do this. This may be the process of the brain development.

The above two points may not cover the concept “autonomous development” completely, but they are the significant features of autonomous developmental network.

Here, two types of network models will be introduced, symbolic and emergent [3]. A symbolic network (SN) means that given a task, it is a human designer in Artificial Intelligence [4], [5] or Cognitive Science [6], [7] who understands the task and handcrafts the model using handpicked task-specific concepts as symbols that the model is only supposed to perform who doesn’t understand the concepts at all. For example, the Finite Automaton (FA) [8] is such a symbolic model. The inputs and states of the FA are both symbolic, designed by human programmers, and the FA itself doesn’t understand. Therefore, a symbolic network is skull open and is not capable of learning like a child. Though a symbolic model may perform well in the specific task, there exist some prominent weaknesses in the natural environments:

• Brittle. Many unpredictable events take place in the external environments. It is impractical for a human designer to guarantee that all the conditions required by the handcrafted model (symbolic model) are met. If the inputs to the network are taken into the consideration during design, the system works well. But if the inputs are unexpected, the system may crash.

• Static. The symbolic model is not able to go beyond the handcrafted representations, it cannot learn or deal with a new task.

• Non-developmental. Although a symbolic representation allows learning, its learning is not fully autonomous. The human teacher selects a learning type. He feeds information into a particular module of the “brain” during training. The network is skull open, not skull closed.

An emergent network means that the internal representations emerge autonomously from network’s interactions with the external world (outside the brain) and the internal world via its sensors and effectors without using the handcrafted content or the handcrafted boundaries for concepts about the extra-body environments.

Fig. 1 illustrates two types of the networks. Obviously, an emergent network is autonomous developmental like our
human brain, while an symbolic network is handcrafted, working well only in the specific task.

Where-What network is a series of brain-inspired emergent network whose features are:

- The WWN is inspired by the dorsal (“where” corresponding to LM) and ventral (“what” corresponding to TM) pathways in cortical visual processing and integrates both bottom-up (feed-forward) and top-down (feedback) connections [9].

- The in-place learning algorithm in WWN is used to develop the internal representations, so that each neuron is responsible for the learning of its own signal processing characteristics within its connected network environment, through interactions with other neurons in the same level. This indicates that the signal processing of each neuron takes place independently [10].

- There are no symbolic representations (handcrafted concepts) in Y area (i.e., closed skull). The internal representations in Y area emerge only via its interactions with sensory end and motor end which are allowed to exposed to the external environment during development. Besides, WWN is not task-specific. Therefore, there is nothing hard-coded to bias the system to do a certain task (e.g., face recognition).

So far, WWN has six versions. WWN-1 [11] can realize object recognition in complex backgrounds performing in two different selective attention modes: the top-down position-based mode finds a particular object given the location information; the top-down object-based mode finds the location of the object given the type. But only 5 locations were tested. WWN-2 [12] can additionally perform in the mode of free-viewing, realizing the visual attention and object recognition without the type or location information and all the pixel locations were tested. WWN-3 [13] can deal with multiple objects in natural backgrounds using arbitrary foreground object contours, not the square contours in WWN-1. WWN-4 used and analyzed multiple internal areas [14]. WWN-5 is capable of detecting and recognizing the objects with different scale in the complex environments [15]. WWN-6 improves the architecture and mechanisms of the network according to the concept “skull closed” [16].

In this paper, the skull closed network WWN-6 is taught to learn a new task, looking for the object in the unknown natural backgrounds and recognizing it like a child. Different from the previous work, the inputs are not artificially synthesized images but image frames drawn from a continuous video taken in the natural environments.

In the remainder of the paper, Section II overviews the architecture and operation of WWN-6. Section III presents some important concepts and algorithms in the network. Experimental results are reported in Section IV. Section V gives the concluding remarks.

II. OVERALL NETWORK SCHEME

In this section, we describe the overall scheme of the network learning and operation.

A. Network Algorithm

Developmental Network (DN) is a generic model about a developmental and computational brain. WWN is a species specialization of DN in the sense that this category of species is such that it has two kinds of motor areas, location motor (LM) and type motor (TM). LM is used for location, representing our pointing and reaching effectors, such as arms and fingers. TM is used for type, representing our vocalization effectors (e.g., vocal tract).

A basic DN, has three areas, the sensory area X, the internal (brain) area Y and the motor area Z. An example of DN is shown in Fig. 2. The internal neurons in Y have bi-directional connection with both X and Z. According to many studies in detailed review by Felleman & Van Essen [17], each area Y
Fig. 3. The structure of WWN-6

(a “bridge”) connects in bi-directionally with many other areas as its two extensive (as its “banks”).

The DP for DNs is task-nonspecific as suggested for the brain in [1] (e.g., not concept-specific or problem specific). In contrast to a static FA, the motor area \( Z \) of a DN can be directly observed by the environment (e.g., by the teacher) and thus can be calibrated through interactive teaching from the environment. The environmental concepts are learned incrementally through interactions with the environments.

The \( X \) area takes video input, but only take one frame at a time. The motor area \( Z \) serves both input and output. When the environment supervises \( Z \), \( Z \) is the input to the network. Otherwise, \( Z \) gives an output vector to drive effectors (muscles) which act on the real world. The order of areas from low to high is: \( X, Y, Z \). For example, \( X \) provides bottom-up input to \( Y \), but \( Z \) gives top-down input to \( Y \).

The most basic function of an area \( Y \) seems to be prediction — predict the signals in its two vast banks \( X \) and \( Y \) through space and time. The prediction applies when part of a bank is not supervised. The prediction also makes its bank less noisy if the bank can generate its own signals (e.g., \( X \)).

The DN algorithm is as follows. Input areas: \( X \) and \( Z \). Output areas: \( X \) and \( Z \). The dimension and representation of \( X \) and \( Y \) areas are hand designed based on the sensors and effectors of the robotic agent or biologically regulated by the genome. \( Y \) is skull-closed inside the brain, not directly accessible by the external world after the birth.

1) At time \( t = 0 \), for each area \( A \) in \( \{X, Y, Z\} \), initialize its adaptive part \( N = (V, G) \) and the response vector \( r \), where \( V \) contains all the synaptic weight vectors and \( G \) stores all the neuronal ages.

2) At time \( t = 1, 2, \ldots \), for each \( A \) in \( \{X, Y, Z\} \) repeat:
   a) Every area \( A \) performs mitosis-equivalent if it is needed, and initialize the weight vector of the new neuron using its bottom-up and top-down inputs \( b \) and \( t \), respectively.
   b) Every area \( A \) computes its area function \( f \), described below,
   \[
   (r', N') = f(b, t, N)
   \]
   where \( r' \) is the new response vector of \( A \).
   c) For every area \( A \) in \( \{X, Y, Z\} \), \( A \) replaces: \( N \leftarrow N' \) and \( r \leftarrow r' \).

In the remaining discussion, \( x \in X \) is always supervised. The \( z \in Z \) is supervised only when the teacher chooses to. Otherwise, \( z \) gives (predicts) effector output.

Put intuitively, like the brain, the DN repeatedly predicts the output \( Z \) for the next moment. When the predicted \( Z \) is mismatched, learning proceeds to learn the new information from \( Z \). But, there is no need to check mismatches: learning takes place anyway.

B. WWN-6 Structure

The network (WWN-6) shown as Fig. 3. Different actions (firing of the neurons) represent different concepts. Here, two categories of concepts are used by the teachers in the external background: the location of the foreground object in the background and the type of the foreground object, corresponding to Location Motor (LM) and Type Motor (TM). Of course, the concepts are not limited to these two (e.g., size of the object).

The \( Y \) area is regarded as the internal brain in the skull, which is off limit to the teachers in the external environment. Using a prescreening area for each source in \( Y \) area, before integration, results in three laminar levels: the ascending level (AL) that prescreenings the bottom-up input, the descending
level (DL) that prescreenings the top-down input and paired level (PL) that combines the outputs of AL and DL. In this model, there exist two pathways and two connections. Dorsal pathway refers to the stream $X := Y := LM$, while ventral pathway refers to $X := Y := TM$, where $\Rightarrow$ indicates that each of the two direction has separate connections.

Two types of connections including forward (bottom-up) and backward (top-down) connections are illustrated by green arrows and blue arrows respectively in Fig. 3.

### III. Concepts and Algorithm Details

#### A. Information Flow among the Areas

To clearly explain, take a simple three-area (one layer in each area) network as an example. For area two, the input of the neuron consists two parts. One is from area one and the other is from area three. Shown as Fig. 2, the connection between area one and area two is local while the connection between area two and area three is global. The output of area two, just the input of the neuron in area three, is the pre-response of the neurons in this area, which indicates that the dimension of the output vector (reshape it to a single vector) equals to the total number of the neurons.

#### B. Pre-response of the Neurons

It is desirable that each brain area uses the same area function $f$, which can develop area specific representation and generate area specific responses. Each area $A$ has a weight vector $\mathbf{v} = (\mathbf{v}_b, \mathbf{v}_t)$. Its pre-response value is:

$$r(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t}) = \hat{\mathbf{v}} \cdot \hat{\mathbf{p}}$$

where $\hat{\mathbf{v}}$ is the unit vector of the normalized synaptic vector $\mathbf{v} = (\mathbf{v}_b, \mathbf{v}_t)$, and $\hat{\mathbf{p}}$ is the unit vector of the normalized input vector $\mathbf{p} = (\mathbf{t}, \mathbf{t})$. The inner product measures the degree of match between these two directions $\hat{\mathbf{v}}$ and $\hat{\mathbf{p}}$, because $r(\mathbf{v}_b, \mathbf{b}, \mathbf{v}_t, \mathbf{t}) = \cos(\theta)$ where $\theta$ is the angle between two unit vectors $\hat{\mathbf{v}}$ and $\hat{\mathbf{p}}$. This enables a match between two vectors of different magnitudes (e.g., a weight vector from an object viewed indoor to match the same object when it is viewed outdoor). The pre-response value ranges in $[-1, 1]$.

In other words, if the synaptic weight vector is considered as the object feature stored in the neuron, the pre-response measures the similarity between the input signal and the object feature.

#### C. Top-k Competition

Top-k competition takes place among the neurons in the same level in $Y$ area, imitating the lateral inhibition which effectively suppresses the weakly matched neurons (measured by the pre-responses). Top-k competition guarantees that different neurons detect different features. The response $r'(t)$ after top-k competition is:

$$r'(t) = \begin{cases} r(t)(r_q - r_{k+1})/(r_1 - r_{k+1}) & \text{if } 1 \leq q \leq k \\ 0 & \text{otherwise} \end{cases}$$

where $r_1$, $r_q$ and $r_{k+1}$ denote the first, $q$th and $(k+1)$th neuron’s pre-response respectively after being sorted in descending order. This means that only the top-$k$ responding neurons can fire while all the other neurons are set to zero.

#### D. How Each $Y$ neuron matches its two input fields

It is important to note that in Fig. 1, teach $Y$ neuron has a limited input field in $X$ but a global input field in $Z$. All $Y$ neurons compete for firing via the above top-k mechanisms. The initial weight vector of each $Y$ neuron is randomly self-assigned, as discussed below. We would like to have all $Y$ neurons to find good vectors in the input space $\{\mathbf{p}\}$. A neuron will fire and update only when its match between $\mathbf{v}$ and $\mathbf{p}$ is among the top, which means that the match for the bottom-up part $\mathbf{v}_b \cdot \mathbf{b}$ and the match for the top-down part $\mathbf{v}_t \cdot \mathbf{t}$ must be both top. Such top matches must be sufficiently often in order for the neuron to mature.

This gives an interesting but extremely important property for attention — relatively very few $Y$ neurons will learn background, since a background patch does not highly correlated with an action in $Z$.

Whether a sensory feature belongs to a foreground or background is defined by whether there is an action that often co-occurs with it.

#### E. ISN and LSN

We need to make each neuron autonomously search in the input space $\{\mathbf{p}\}$ but keep its age at 0 until its pre-response value is sufficiently large to indicate that its current feature vector is meaningful (instead of garbage-like). A garbage-like vector cannot converge to a desirable target based on Hebbian learning. Hebbian learning should not kick in until the vector is within the signal manifold.

Therefore, there exist two types of neurons in the $Y$ area (brain), initial state neurons (ISN) and learning state neurons (LSN). After the initialization of the network, all the neurons are in the initial state. During the learning of the network, neurons may be transformed from initial state into learning state, which is determined by the pre-response of the neurons. In our network, a parameter $\epsilon_1$ is defined. If the pre-response is over $1 - \epsilon_1$, the neuron is transformed into learning state, otherwise, the neuron keeps the previous state.

#### F. Hebbian-like Learning

Hebbian-like learning is described as:

$$v_j(t + 1) = w_1(n(t))v_j(t) + w_2(n(t))r'(t)\hat{p}$$

where $r'(t)$ is the response of the neuron after top-k competition, $w_1$ and $w_2$ are the two parameters representing retention rate and learning rate with $w_1 + w_2 \equiv 1$. These two parameters are defined as following:

$$w_1(n) = 1 - w_2(n), \quad w_2(n) = \frac{1 + u(n)}{n}$$
where $u(n)$ is the amnesic function:

$$u(n) = \begin{cases} 0 & \text{if } n \leq t_1 \\ c(n - t_1)/(t_2 - t_1) & \text{if } t_1 < n \leq t_2 \\ c + (n - t_2)/r & \text{if } t_2 < n \end{cases}$$

where $n$ is the firing age of the neuron, $t_1 = 20, t_2 = 200, c = 2, r = 10000$.[18]

Only the firing neurons (firing neurons are in learning state definitely) and all the neurons in initial state will implement Hebbian-like learning, updating the synaptic weights according to the above formulas. In $Y$ area, if the neuron in learning state is one of the top-$k$ winners and its pre-response is over $1 - \epsilon_2$, the neuron will be fired and implement Hebbian-like learning. The firing age of the neurons in learning state and initial state is updates as

$$n(t+1) = \begin{cases} n(t) & \text{if the neuron is ISN} \\ n(t) + 1 & \text{if the neuron is top-$k$ LSN.} \end{cases}$$

To a neuron, the lower the firing age the higher the learning rate. That is to say, ISN is more capable to learn new concepts than LSN. If the neurons are regarded as resources, ISNs are the idle resources while LSNs are the developed resources. So, the resources utilization (RU) in $Y$ area can be calculates as

$$RU = \frac{N_{LSN}}{N_{LSN} + N_{ISN}} \times 100\%$$

where RU represents the resources utilization, $N_{LSN}$ and $N_{ISN}$ are the number of LSN and ISN.

G. Pulvinar

Pulvinar is an area inside the thalamus believed to play a role in attention [19]. The pulvinar module exists in WWN-1 to 5, which suppresses the neurons whose receptive fields are outside the current object being learned. This indicates that the neuron (or its neighbors) covering the object being learned is always the firing neuron (winner neuron) with the pulvinar module. But, the pulvinar module used in WWN-1 through WWN-5 is pre-programmed, not consistent with the concept "skull closed" as much as we liked, since pulvinar assumes that there is an early developed internal brain area pulvinar that represents location although in an approximate way.

In this developmental network, there does not exist the pulvinar module. The new guiding concept that the actions of the network are related to some patches (foreground object) in $X$ area is used in our new developmental model, making the model more and more mature during the training even without pulvinar module.

H. Training and Testing

The architecture of the network is the same during the training and testing. During training, the network is supervised
by the external teacher. But for testing, there is no imposed action, i.e., no teacher supervision. The trained network should provide the correct action for a given test sample drawn from the same distribution as the training data, even if the test sample is not specifically taught.

IV. EXPERIMENTS AND RESULTS

In order to imitate the cognitive learning of a little child, the training of the network should be as natural as possible. In our experiment, 24 objects are used to be learned which are shown as Fig. 4. Though the objects displayed here are all animals, the network is not limited to these.

The raw video clips of each object to be learned were completely taken in the real natural environments. During video capture, the object held by the teacher’s hand was required to move slowly so that the agent could pay attention to the object. Fig. 5 shows the example frames extracted from a continuous video clip as an illustration. For the child, the teacher will provide the correct information of the samples. In our experiments, the prior information includes the type and the location of the object in the unspecific natural backgrounds. Therefore, before the training of the network, the raw video clips need to be preprocessed, automatically or semi-automatically, to provide the approximate location of the object in the background.

Cross validation are applied for the training and the testing of the developmental network: in the first round, even frames for training and odd frames for testing, and then in the second round, odd frames for training and even frames for testing. Thus, for the agent in the training phase, the inputs are the natural video frames just like the scene that a teacher holds an object and moves it slowly, telling the child what it is and where it is. After the training, usually the teacher will check whether the child has learnt the objects well, which is similar to the testing phase of the network in the experiment.

The performance of the network, type recognition rate and location error, is shown as Fig. 7. In each subfigre, three performance curves are drawn which correspond to three different attention modes respectively. In free-viewing mode, the agent looks for and recognizes any learnt object in the unknown natural backgrounds without top-down attention, which is like the question, “what kind of object have you seen in the image and where is it?”; Type-based attention means that the agent works with the top-down attention based on the type of the object, corresponding to the teacher’s question, “where is the cat (the type of the specific object)?”. Similarly, location-based attention means that the agent need to recognize the object at specific location told by the teacher, like “what is the object on the top left?”. From Fig. 7, the experimental results show that the WWN-6 developmental network can learn objects like a child in a natural way, i.e., via the natural scenes. Furthermore, the experimental results indicate that either the type-based attention or the location-based attention can help the agent perform the object recognition better.

In order to study the details of WWN-6 networks after learning the specific objects in the natural video frames, the synaptic weights of the neurons in Y area are visualized in Fig. 6. It shows that every neuron in each depth detects a specific object (“what”) feature in a specific position (“where”). Due to the limited neuronal resource in Y area, some neurons deal with one or multiple objects at multiple pixel locations.

Fig. 8 illustrates the resource utilization in 10 epochs, from 85% in the first epoch to 100% in the sixth epoch, which is the development of the agent like our developmental brain.

V. CONCLUSION

According to the theory and experiments presented in the paper, the frequency of a background patch does not have to be low in order for the brain to learn well. It is the action from the brain that determine whether an input patch is attended or not. WWN-6 has learnt to detect and recognize the general objects with curve contours in the unknown natural environments. This is a computational prediction on how a human brain autonomously decides which sensory patch to attend among many. WWN-6 training demonstrated that motor-supervised teaching can effectively identify such object patches from unknown complex natural backgrounds.

In the future work, more objects with different views will be learnt. And the synapse maintenance and the cell regenesis will be included into the network.

REFERENCES

Fig. 5. Frames extracted from a continuous video clip and used in the training and testing of the network

Fig. 6. Visualization of the weights in one depth in $Y$ area (ten depths totally), which have three types: bottom-up weights (connected from $X$ area), top-down weights (connected from TM) and top-down weights (connected from LM). Block color in (b) represents the type of the specific object, and the colors corresponding to 24 objects are mapped into a color bar with a value range from 0 to 1.

Fig. 7. Network performance variation within 10 epochs under three different attention modes: free-viewing mode, type-based mode and location-based mode. One epoch of training means that the agent is taught an object with all the possible locations in the backgrounds. The depth in $Y$ area of our network is set 10, meaning that for 24 object to be learned, the resource shortage is $(24 - 10)/24 = 58.3\%$. If the depth is 24, indicating that each neuron corresponds to one object of one location, the resource is enough without considering the various backgrounds.